Journal of *Marine Science and Engineering*

MDPI

Article

Three-Dimensional Dynamic Trajectory Planning for Autonomous Underwater Robots Under the PPO-IIFDS Framework

Liqiang Liu [1], Min Sun [1,*], Enjiao Zhao [2] and Kuang Zhu [1]

[1] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; liuliqiang@hatrannavi.com (L.L.); 212308802004@zust.edu.cn (K.Z.)
[2] College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; zhaoenjiao935@hrbeu.edu.cn
* Correspondence: 222308855053@zust.edu.cn

Abstract: Three-dimensional (3D) dynamic trajectory planning for Autonomous Underwater Vehicles (AUVs) is associated with significant challenges such as balancing the trajectory quality, computational efficiency, and environmental adaptability within complex dynamic environments. To tackle these challenges, this paper proposes a novel trajectory planning framework by integrating Proximal Policy Optimization (PPO) and an Improved Interfered Fluid Dynamic System (IIFDS). The IIFDS serves as the planning layer, generating obstacle-adaptive trajectories for AUVs through the dynamic adjustment of flow field parameters. Meanwhile, PPO functions as the learning and decision-making layer, optimizing critical parameters in IIFDS, including repulsion response coefficients, tangential response coefficients, and directional coefficients, to enhance adaptability and real-time decision-making. To meet specific mission requirements, the IIFDS incorporates dynamics and kinematics constraints, while the PPO reward function is improved with a multi-objective dynamic structure. This reward design integrates objectives such as obstacle avoidance, target distance minimization, trajectory smoothness, dynamics constraints, and energy efficiency. These enhancements address sparse reward issues effectively and significantly improve the convergence and practical applicability of trajectory planning. Additionally, a diverse and dynamically complex obstacle environment is constructed for model training and performance evaluation. The experimental results demonstrate that the proposed framework efficiently generates smooth, energy-efficient, and collision-free trajectories in high-density dynamic obstacle scenarios. The framework exhibits strong robustness, excellent generalization capabilities, and offers a reliable solution for 3D dynamic trajectory planning for AUVs.

Keywords: autonomous underwater vehicles (AUVs); trajectory planning; improved interfered fluid dynamic system (IIFDS); proximal policy optimization (PPO)

Academic Editor: Rafael Morales

Received: 27 January 2025
Revised: 17 February 2025
Accepted: 24 February 2025
Published: 26 February 2025

Citation: Liu, L.; Sun, M.; Zhao, E.; Zhu, K. Three-Dimensional Dynamic Trajectory Planning for Autonomous Underwater Robots Under the PPO-IIFDS Framework. *J. Mar. Sci. Eng.* **2025**, *13*, 445. https://doi.org/10.3390/jmse13030445

1. Introduction

The increasing global demand for marine resource exploration and environmental protection has driven the widespread application of Autonomous Underwater Vehicles (AUVs) in fields such as deep-sea exploration, seabed topographic mapping, environmental monitoring, and resource prospecting [1]. However, the complexity and dynamic nature of the marine environment—characterized by variable ocean currents, dense moving obstacles (such as ice floes or marine organisms), and irregular three-dimensional terrains—poses significant challenges to AUV navigation. Consequently, achieving efficient and reliable 3D

trajectory planning for AUV in dynamic and complex environments has become a crucial research topic in ocean engineering and intelligent control.

Existing studies show that AUV trajectory planning tasks encompass three major categories of algorithms: classical graph search algorithms, heuristic algorithms, and intelligent algorithms. Each category has its own strengths and limitations and is suitable for different environmental conditions and performance requirements based on the nature of the problem and the application scenario. Specifically, classical algorithms, such as A* [2] and Dijkstra [3], are based on rigorous mathematical theory and are capable of finding optimal paths in known and stable static environments. These algorithms are computationally efficient and well-suited for trajectory planning problems. However, they struggle to handle dynamic obstacles, high-dimensional complex environments, and the computational burdens of large-scale problems. Heuristic algorithms, such as the Artificial Potential Field (APF) [4] method and Rapidly Exploring Random Trees (RRTs) [5], incorporate heuristic functions or random extension strategies. These methods exhibit good real-time responsiveness and flexibility in dynamic and complex environments, enabling rapid obstacle avoidance and efficient pathfinding. However, they are prone to local optima and heavily rely on parameter tuning, which limits their ability to guarantee globally optimal solutions. Intelligent algorithms, such as Deep Reinforcement Learning (DRL) [6] and Genetic Algorithms (GAs) [7], mimic biological learning mechanisms and are well suited to handling complex and dynamically changing environments. They exhibit strong adaptability and self-learning capabilities, especially for multi-objective optimization problems. Nevertheless, the training process for intelligent algorithms requires large amounts of data and computational resources, and their results often lack stability and convergence, particularly in highly uncertain environments. In practical applications, these three categories of algorithms have different applications depending on the task characteristics and performance requirements: classical algorithms are suited for optimal pathfinding in known environments, heuristic algorithms are ideal for real-time reactions in dynamic environments, while intelligent algorithms excel at tackling highly uncertain, complex, and multi-objective dynamic tasks.

The APF method is a commonly used local planning approach, characterized by its simplicity, low computational complexity, and fast responsiveness, making it suitable for online planning. However, traditional APF methods may fail to generate feasible or optimal paths due to issues such as target inaccessibility and local optima. To address this, many researchers have analyzed and improved the APF method. One notable example is the work of Ge Shuzhi's team at the National University of Singapore [8], which enhanced the repulsive potential field function to address issues such as local minima, unreachable targets, and the avoidance of moving threats. Despite these improvements, APF still lacks the concept of obstacle shapes (envelopes) and relies entirely on force field adjustments to generate paths. As a result, improper parameter tuning may cause AUVs to enter obstacles, leading to failures in obstacle avoidance. To overcome the limitations of APF, researchers proposed the stream function method based on the fundamental principles of potential fields [9,10]. This method provides advantages such as a fast planning speed and smooth trajectories. However, the concept of stream functions becomes invalid when extending it from two-dimensional to three-dimensional planning spaces, limiting this method to 2D trajectory planning.

To address this issue, researchers introduced a 3D trajectory planning method inspired by the "flowing water avoiding stones" principle [11], which references the macroscopic behavior of natural water flow: water flows in a straight line in the absence of obstacles, while it smoothly bypasses obstacles and continues toward its target when obstructions are present. This method integrates trajectory planning with fluid computation by introducing

the concept of three-dimensional obstacle envelopes. However, traditional flow-based methods still have significant limitations: (1) analytical methods can only handle spherical obstacles; (2) due to the need for computational fluid dynamics simulations, the computational cost is excessively high, restricting these methods to offline trajectory planning.

To address the limitations of traditional flow-based methods, the Interfered Fluid Dynamical System (IFDS) algorithm was first proposed [11]. Based on analytical methods, IFDS avoids solving fluid equations with complex boundary conditions, making it suitable for handling complex terrains and various obstacle shapes. The planned routes not only retain the natural characteristics of flow-based methods but also feature simple environmental modeling and a low computational cost, significantly expanding the applicability of flow-based methods. However, the streamline distribution generated by IFDS has certain limitations and is prone to local traps and stagnation points, which cannot be fundamentally resolved by auxiliary strategies alone [12]. The root cause of these issues lies in the insufficiently objective and comprehensive definition of the perturbation matrix, which limits the spatial distribution of streamlines. To address this, the Improved Interfered Fluid Dynamical System (IIFDS) algorithm was proposed [13]. By introducing a tangential matrix into the perturbation matrix, IIFDS effectively addresses these limitations. Compared to IFDS, IIFDS redefines the perturbation matrix by incorporating tangential velocity components into the perturbation flow, allowing it to point in any direction. By adjusting the repulsion response coefficients in the repulsion matrix and the tangential response coefficients and directional coefficients in the tangential matrix, IIFDS generates a variety of streamline shapes distributed throughout the planning space. These streamlines are then filtered to select paths that avoid local traps and stagnation points. However, some of these streamlines fail to meet AUV dynamics constraints or incur excessively high trajectory costs. Therefore, it is necessary to optimize the coefficients to select a trajectory that satisfies environmental and kinematic constraints while ensuring optimal performance under specific metrics or multiple objectives.

In recent years, new-generation artificial intelligence methods represented by DRL have been widely applied to the optimization and control of complex systems. These machine learning methods have several advantages [14–16]: (1) they do not rely on environmental models or prior knowledge, and policies can be improved solely through interaction with the environment; (2) the deep neural networks used in DRL have powerful nonlinear approximation capabilities, making them effective for optimizing high-dimensional continuous state-action spaces, which is fundamental to 3D trajectory planning in complex dynamic environments; and (3) the policies obtained through DRL require only a forward pass during inference, making them highly suitable for decision-making tasks with high real-time requirements. Based on these advantages, some researchers have explored the application of DRL in planning. For example, [17] proposed an end-to-end perception–planning–execution framework based on a two-layer deterministic policy gradient algorithm to address challenges related to training and learning in end-to-end control approaches. Similarly, [18] proposed an online collision avoidance planning algorithm based on active sonar sensors for obstacle detection. While these methods achieve good planning results, three key issues warrant further investigation:

First, DRL, as a general decision-making framework, may struggle to simultaneously ensure safety and trajectory smoothness when addressing the specific problem of AUV 3D dynamic trajectory planning. The simulation results indicate that directly using DRL to generate control inputs for trajectory planning ensures fast and safe obstacle avoidance but often produces trajectories lacking smoothness, which hinders precise tracking by low-level controllers. Combining DRL with classical heuristic methods could leverage their respective optimization speed and trajectory quality, leading to improved planning

results. However, designing a hybrid framework that effectively handles complex dynamic obstacles (e.g., 3D obstacles with varying sizes and trajectories) remains a challenge.

Second, DRL-based trajectory planning methods require agents to interact with simulated task environments and update the weights of deep neural networks based on environmental feedback. The trained deep action networks are then deployed for online planning in real-world environments. Therefore, designing simulation environments tailored to the trajectory planning methods being used is essential for improving training efficiency and ensuring the policy's generalization in complex obstacle scenarios. Unfortunately, existing studies lack targeted research on systematic modeling methods for training environments.

Finally, high-quality trajectories must consider multiple objectives simultaneously, such as obstacle avoidance effectiveness, target reachability, trajectory smoothness, energy consumption within acceptable ranges, and adherence to AUV dynamics and kinematics constraints. Most current studies focus on single-objective optimization, which does not align with the practical requirements of AUV trajectory planning.

Contributions of This Paper:

(1) A trajectory planning framework integrating PPO and IIFDS:

This paper designs a 3D dynamic trajectory planning framework for AUVs, integrating PPO with the IIFDS. In this framework, IIFDS serves as the planning layer, dynamically adjusting the flow field parameters to generate obstacle-adaptive trajectories in dynamic environments. PPO acts as the learning and decision-making layer, optimizing the flow field disturbance parameters and dynamically adjusting planning strategies, enabling efficient coordination between the two algorithms in dynamic obstacle environments.

(2) Key improvements to the PPO and IIFDS algorithms:

PPO algorithm improvement: For the trajectory planning task, a multi-objective dynamic reward function is designed, incorporating obstacle avoidance, target distance, trajectory smoothness, dynamic constraints, and energy consumption. This approach effectively addresses the sparse reward problem in traditional methods, significantly improving the algorithm's convergence and the practicality of trajectory planning.

IIFDS algorithm improvement: Task-specific dynamic and kinematic constraints for AUVs are introduced into the IIFDS planning layer. This ensures that the generated trajectories not only satisfy environmental constraints but are also executable, enhancing the reliability and applicability of the planning results in real-world scenarios.

(3) Construction of a dynamic and complex obstacle environment for model training and framework validation:

A diverse dynamic obstacle environment model is developed, capable of simulating various obstacle behaviors and complex scenarios. This environment supports interactions between the agent and the environment while dynamically generating training data of varying complexity. As a result, it improves training efficiency and enhances the framework's generalization performance in practical applications. During testing, the environment validates the framework's trajectory planning performance in high-density dynamic obstacle scenarios, including the generation of collision-free trajectories, trajectory smoothness, and energy efficiency.

2. A Three-Dimensional Dynamic Trajectory Planning Framework Based on PPO and IIFDS

To simplify the analysis in this paper, the following assumptions are made:

Assumption 1. *All obstacles, including both static and dynamic obstacles, are approximated as standard convex polyhedrons, such as spheres, ellipsoids, and cylindrical shapes. The mathematical representation of an obstacle's boundary can be defined as follows:*

$$\phi_k(P) = \left(\frac{x - x_0}{a}\right)^{2p} + \left(\frac{y - y_0}{b}\right)^{2q} + \left(\frac{z - z_0}{c}\right)^{2r} - 1 \tag{1}$$

where $\phi_k(P)$ represents an implicit expression for the boundary of an obstacle. A value of 0 indicates that point P lies exactly on the obstacle's surface, a value less than 0 indicates that P is inside the obstacle, and a value greater than 0 indicates that P is outside the obstacle. $(x, y, z)$are the coordinates of point$P$, while (x_0, y_0, z_0) denote the coordinates of the obstacle's center in the three-dimensional space. a, b, ccontrol the extent of the obstacle along the$x$, y, and z directions (i.e., the size of the obstacle). p, q, rcontrol the curvature of the obstacle along each direction (i.e., the exponents of the shape parameters).

Assumption 2. *The information on the target location and obstacle status at the current moment is available online. This information includes, but is not limited to, the position and velocity.*

2.1. Introduction to the Improved Interfered Fluid Dynamical System (IIFDS)

2.1.1. Initial Flow Field Velocity Model

The IIFDS algorithm is based on mimicking the characteristics of natural fluid dynamics. Under conditions free from interference, the initial flow field streamlines are directed straight toward the target point. The model of the initial flow field is shown in Figure 1.

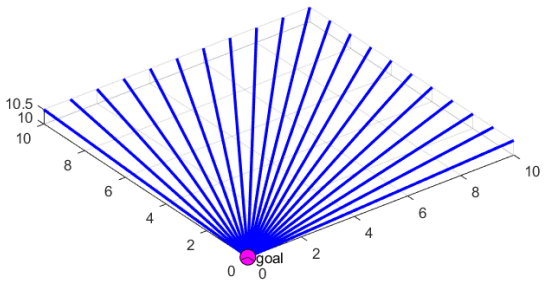

Figure 1. Initial flow field model.

In the initial flow field, the velocity vector of the fluid passing through each position P, denoted as $U(P)$, can be expressed as follows:

$$U(P) = -\frac{V_0}{d(P, P_d)}(P - P_d) \tag{2}$$

where $U(P)$ represents the velocity vector of the initial flow field at the current position of the AUV. V_0 is a virtual velocity constant, used to determine the intensity of the flow field. $P = (x, y, z)$ denotes the current position of the AUV, $P_d = (x_d, y_d, z_d)$ represents the target position, and $d(P, P_d)$ denotes the Euclidean distance between the current position P and the target point P_d, which is given by the following:

$$d(P, P_d) = \sqrt{(x - x_d)^2 + (y - y_d)^2 + (z - z_d)^2} \tag{3}$$

2.1.2. Obstacle Influence Modeling

When obstacles exist in the environment, their influence on the predefined initial flow field changes the fluid's velocity direction, resulting in flow field distortion. The effect of obstacles on the initial flow field is modeled using an influence matrix, defined as follows:

$$M(P) = \sum_{k=1}^{K} \omega_k(P) M_k(P) \tag{4}$$

where $M(P)$ represents the overall disturbance matrix, $\omega_k(P)$ denotes the weight coefficient of the k-th obstacle, $M_k(P)$ is the disturbance matrix of the k-th obstacle, and K indicates the total number of obstacles.

The weight coefficient for each obstacle is defined as follows:

$$\omega_k(P) = \begin{cases} 1, K = 1, \\ \prod\limits_{i=1, i \neq k}^{K} \dfrac{\phi_i(P) - 1}{(\phi_i(P) - 1) + (\phi_k(P) - 1)}, K \neq 1 \end{cases} \tag{5}$$

where $\phi_k(P)$ represents the implicit equation of the obstacle's surface, defining whether P is inside, on, or outside the obstacle.

The influence matrix $M_k(P)$ of the k-th obstacle is defined based on repulsion and redirection effects, as follows:

$$M_k(P) = I - \frac{n_k n_k^T}{\left| n_k^T n_k \right| \cdot T^{\frac{1}{\rho_k}}} + \frac{t_k n_k^T}{|t_k| \cdot |n_k| \cdot T^{\frac{1}{\sigma_k}}} \tag{6}$$

where I denotes the 3×3 identity matrix, also referred to as the attraction matrix, which functions similarly to the attractive force in the artificial potential field method. n_k represents the normal vector of the obstacle's surface, t_k represents the tangential vector, and ρ_k is the repulsion coefficient that controls the intensity of the obstacle's repulsive effect. A larger ρ_k value enables the disturbed fluid to avoid obstacles in the environment earlier. σ_k denotes the tangential response coefficient, which regulates the intensity of tangential fluid flow. T represents the distance from the obstacle's center to the current position, normalized by the obstacle's radius.

In a three-dimensional environment with obstacles, the presence of obstacles causes deviations in the original flow field paths, resulting in a disturbed flow field. This disturbed flow field is capable of avoiding obstacles while converging toward the target point. The disturbed flow field model is illustrated in Figure 2.

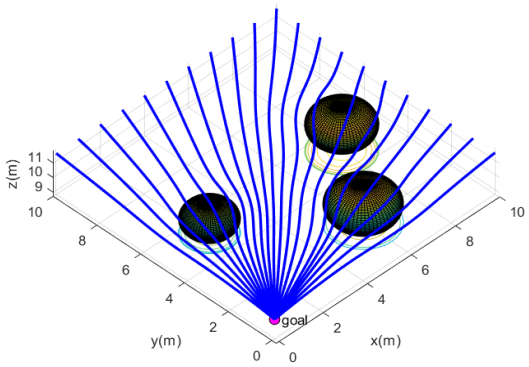

Figure 2. Disturbed flow field model.

2.1.3. Tangential Vector Modeling

In the practical application of the IIFDS algorithm, the flow lines generated for obstacle avoidance are often restricted to a single plane, which may result in trajectories becoming trapped in local minima or stagnating at certain points. To address this issue, the IIFDS algorithm introduces the concept of a tangential matrix, allowing flow lines to move in arbitrary directions around obstacles rather than being confined to a single plane. When approaching stagnation points, the IIFDS algorithm enhances the tangential matrix to provide tangential momentum along the obstacle surface, effectively preventing the AUV from lingering at stagnation points.

On the tangent plane defined by the normal vector n_k, any tangential vector t_k is generated as follows:

$$t_k = R_k t'_k \tag{7}$$

where t'_k represents the tangential vector in the local tangential coordinate system, determined based on the tangential angle θ. R_k denotes the rotation matrix that transforms the local tangential basis vectors into the global coordinate system, defined as follows:

$$R_k = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

In this context, θ controls the rotational angle of the tangential direction, determining the specific direction for bypassing the obstacle.

2.1.4. Influence of Dynamic Obstacles

The velocity of dynamic obstacles affects the flow field through the following equation:

$$v_{obs} = e^{-\frac{T}{\lambda}} \cdot V_{obs} \tag{9}$$

where v_{obs} represents the velocity influence of the dynamic obstacle at the current position. T denotes the normalized distance between the current point and the obstacle. λ is the attenuation factor, which controls the influence range of the obstacle's velocity on the flow field. V_{obs} represents the velocity vector of the dynamic obstacle.

2.1.5. Comprehensive Velocity Calculation

The comprehensive velocity of the AUV is calculated using the following equation:

$$\overline{U}(P) = M(P) \cdot U(P) - v_{obs} \tag{10}$$

2.1.6. Trajectory Update

The next trajectory point is determined by integrating the velocity $\overline{U}(P)$, according to the following equation:

$$\{P\}_{i+1} = \{P\}_i + \overline{U}(P)\Delta t \tag{11}$$

where Δt is the time step.

2.1.7. Heading and Pitch Angle Constraints

From the above derivation, it can be seen that traditional IIFDS does not explicitly consider the motion model and constraints of the AUV during trajectory planning. In 3D dynamic trajectory planning for AUVs, to ensure that the generated trajectory satisfies both the optimization objectives of flow-field perturbation and the motion capability constraints of the AUV itself, we introduce dynamics and kinematics constraints. These constraints

primarily act on the updates to the heading angle and pitch angle, and by adjusting the next position, they ensure the physical feasibility of the trajectory.

Based on the current position of the AUV, $P_i = (x_i, y_i, z_i)$, the position at the previous time step, $P_{i-1} = (x_{i-1}, y_{i-1}, z_{i-1})$, and the next position predicted by the flow field, $P_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1})$, the heading angle and its variation are calculated as follows:

$$\Delta \Psi = \Psi_{i+1} - \Psi_i \tag{12}$$

where Ψ_i denotes the heading angle from P_{i-1} to P_i, and Ψ_{i+1} represents the heading angle from P_i to P_{i+1}.

If $|\Delta\Psi| > \Psi_{\max}$, the heading angle is corrected as follows:

$$\Psi_{re} = \Psi_i + \text{sign}(\Delta\Psi) \cdot \Psi_{max} \tag{13}$$

The correction logic for the pitch angle is analogous to that of the heading angle and is not repeated here. After completing the corrections for the heading and pitch angles, the previously planned position for the next time step is updated accordingly. The corrected position is given by the following:

$$\widetilde{P}_{i+1} = P_i + \Delta s \cdot \begin{bmatrix} \cos(\gamma_{re}) \cdot \cos(\Psi_{re}) \\ \cos(\gamma_{re}) \cdot \sin(\Psi_{re}) \\ \sin(\gamma_{re}) \end{bmatrix} \tag{14}$$

where $\widetilde{P}_{i+1} = \left(\widetilde{x}_{i+1}, \widetilde{y}_{i+1}, \widetilde{z}_{i+1}\right)$ represents the next position of the AUV after incorporating the dynamics and kinematics constraints, and γ_{re} denotes the corrected pitch angle.

2.2. Introduction to the Improved PPO Algorithm

From Equation (6), it can be seen that the influence matrix $M_k(P)$ is not only related to the position of the AUV and the implicit equation of the obstacle's surface, but also to the repulsion coefficient ρ_k, tangential response coefficient σ_k, and directional coefficient θ_k of each obstacle. If these three parameters are fixed, the resulting trajectory may fail to meet the specific requirements of certain scenarios or obstacles, leading to suboptimal trajectory planning.

As shown in Figure 3, the plotted trajectories represent different parameter combinations. The orange trajectory in the bottom-right corner corresponds to $\rho = 0.5$, $\sigma = 0.5$, and $\theta = 0$. From the bottom-left to the top-left corner, the coefficients increase by 0.2 for each trajectory. Different parameter combinations determine the shape and direction of the trajectories. In previous research [19], receding horizon control (RHC) has been used to optimize these parameters online. However, the serial nature of RHC's solution mechanism is not well suited to the robustness and real-time requirements of complex dynamic obstacle environments.

PPO is a policy-based deep reinforcement learning algorithm and an off-policy algorithm. With its high stability and strong real-time performance in complex dynamic environments, we choose the PPO algorithm to optimize the three parameters of the IIFDS algorithm: the repulsion coefficient ρ_k, the tangential coefficient σ_k, and the directional coefficient θ_k. The PPO algorithm comprises five core components: environment, agent, state, reward function, and action. The following sections will provide detailed explanations of these five components.

In the 3D dynamic trajectory planning task, the environment is a three-dimensional space that contains multiple dynamic spherical obstacles with varying radii and trajectories. In this environment, it is possible to observe the instantaneous velocities of the dynamic

obstacles at any given moment, the distance between the AUV and the surface of the obstacles, as well as the relative position of the AUV to the target point. This dynamic environment simulates various complex scenarios that the AUV might encounter during real-world navigation, providing a realistic basis for testing and validating the algorithm.

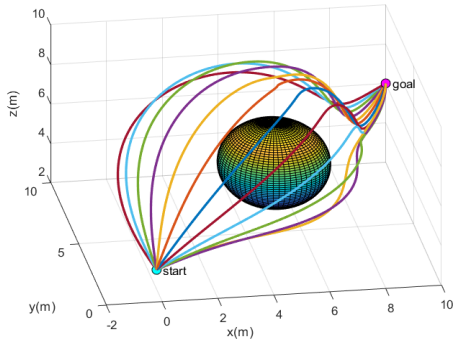

Figure 3. The impact of different reaction coefficients and directional coefficient combinations on planned trajectories.

In the 3D dynamic trajectory planning task, the AUV is considered as the agent.

The state space represents the collection of environmental information perceived by the agent. It comprises three vectors: the vector pointing to the target point, the vector pointing to the surface of the nearest obstacle, and the velocity vector of the nearest obstacle.

In traditional deep reinforcement learning methods, the reward function in the PPO algorithm often suffers from the issue of sparse rewards, which makes the learning process less adaptive to specific tasks. This issue is particularly pronounced in the 3D dynamic trajectory planning task for AUVs, where traditional reward designs fail to effectively guide the AUV in performing fine-grained action control. To address this, the paper introduces an improved reward function aimed at providing more precise guidance for the dynamic trajectory planning of the AUV. This reward function comprises five main components: obstacle avoidance reward, target distance reward, trajectory smoothness reward, dynamics constraint reward, and energy consumption reward.

(1) Obstacle Avoidance Reward

The obstacle avoidance reward encourages the AUV to remain a safe distance from obstacles while considering the velocity of the obstacles. The reward is calculated as follows:

$$R_{\text{avoid}} = \begin{cases} d_{\text{factor}}\left(\frac{d_{\text{toobs}} - R_{\text{obs}}}{R_{\text{obs}}} - 1\right), if\ d_{\text{toobs}} \le R_{\text{obs}} \\ d_{\text{factor}}\left(\frac{d_{\text{toobs}} - R_{\text{th}}}{R_{\text{th}}} - 0.3\right), if\ R_{\text{obs}} < d_{\text{toobs}} \le R_{\text{th}} \end{cases} \tag{15}$$

where $d_{\ factor} = 1 + 0.5 * \left|\vec{v}_{obs}\right|$ represents the dynamic influence factor of the obstacle's velocity on the reward. Faster obstacle velocities increase the dynamic impact factor, which amplifies the reward's sensitivity to obstacle avoidance. d_{toobs} is the distance between the AUV and the center of the obstacle. R_{obs} is the radius of the obstacle. R_{th} is the threshold radius of the obstacle's influence zone. $\vec{v}_{obs}$ is the velocity vector of the obstacle.

(2) Target Distance Reward

The target distance reward encourages the AUV to move closer to the target point, with the reward increasing as it approaches the target. The reward is calculated as follows:

$$R_{\text{goal}} = -\frac{\log(d_{\text{togoal}} + \epsilon)}{\log(d_{\text{total}} + \epsilon)} \tag{16}$$

where d_{togoal} represents the distance between the current position of the AUV and the target point. d_{total} represents the total distance from the starting point to the target point.

ϵ is a small positive constant added to prevent logarithmic computation errors.

If the AUV is very close to the target (less than a specified threshold value), an additional bonus reward is given:

$$\text{if} d_{\text{togoal}} \leq \text{threshold}, R_{\text{goal}} += 5 \tag{17}$$

(3) Trajectory Smoothness Reward

The trajectory smoothness reward is designed to encourage the AUV to move along a smooth trajectory, avoiding abrupt changes in direction, velocity, or acceleration. The reward is defined as follows:

$$R_{\text{smooth}} = -\frac{|\phi|}{2\pi} \cdot 0.5 - \lambda_1 \cdot \Delta v - \lambda_2 \cdot \left| \vec{a} \right| \tag{18}$$

where ϕ represents the angle between the current motion direction of the AUV and the direction toward the target. Δv denotes the change in velocity between the current and previous time steps. $\vec{a}$ represents the current acceleration of the AUV. λ_1 and λ_2 are weighting factors penalizing changes in velocity and acceleration, respectively.

(4) Kinematic Constraint Reward

The kinematic constraint reward ensures that the AUV's motion adheres to kinematic limits, avoiding excessive heading angles, pitch angles, and over-speeding behaviors. The reward is defined as follows:

$$R_{\text{kinematic}} = -\left(\frac{|x_1 - \Psi_{\text{re}}|}{\Psi_{\max}} + \frac{|z_1 - \gamma_{\text{re}}|}{\gamma_{\max}} \right) - \lambda_3 \cdot \left(\left| \vec{v} \right| - v_{\max} \right) - \lambda_4 \cdot \left(\left| \vec{a} \right| - a_{\max} \right) \tag{19}$$

where x_1 andz_1 represent the AUV's current heading and pitch angles. Ψ_{re} and γ_{re} represent the corrected heading and pitch angles. $\Psi_{\max}$ and $\gamma_{\max}$ denote the maximum allowable variations in the heading and pitch angles. $\left| \vec{v} \right|$ represents the current speed of the AUV, and $\left| \vec{a} \right|$ represents the current acceleration of the AUV. $v_{\max}$ and $a_{\max}$ denote the maximum allowable speed and acceleration of the AUV. λ_3 and λ_4 are weighting factors that penalize excessive speed and acceleration, respectively.

(5) Energy Consumption Reward

This reward penalizes energy-inefficient behaviors and encourages the AUV to adopt energy-saving motion strategies. It is defined as follows:

$$R_{\text{energy}} = -\lambda_5 \cdot \left| \vec{v} \right|^2 \tag{20}$$

where $\left| \vec{v} \right|$ represents the current speed of the AUV. λ_5 is the weighting factor for the energy consumption penalty.

(6) Total Reward Function

By combining all the individual reward components, the total reward function is obtained, which comprehensively guides the AUV to accomplish its tasks while ensuring stable, safe, and efficient motion. The total reward is expressed as follows:

$$R_{\text{total}} = R_{\text{avoid}} + R_{\text{goal}} + R_{\text{smooth}} + R_{\text{kinematic}} + R_{\text{energy}} \tag{21}$$

The trajectories generated by the IIFDS algorithm are determined by three parameters within the algorithm: the repulsion coefficient ρ_k, the tangential coefficient σ_k, and the directional coefficient θ_k. Therefore, we select these three parameters as the action outputs of the improved PPO algorithm.

2.3. Optimization of IIFDS Parameters Using the Improved PPO

To enable trajectory planning for AUVs in complex dynamic three-dimensional environments, this paper proposes a trajectory planning framework based on the integration of PPO and IIFDS. In this framework, IIFDS serves as the planning layer, adjusting the flow field parameters dynamically to generate trajectories that adapt to moving obstacles. PPO acts as the learning and decision-making layer, optimizing the IIFDS flow field parameters, including the repulsion coefficient ρ_k, the tangential coefficient σ_k, and the directional coefficient θ_k. The detailed process of the integrated algorithm is shown in the flowchart in Figure 4.

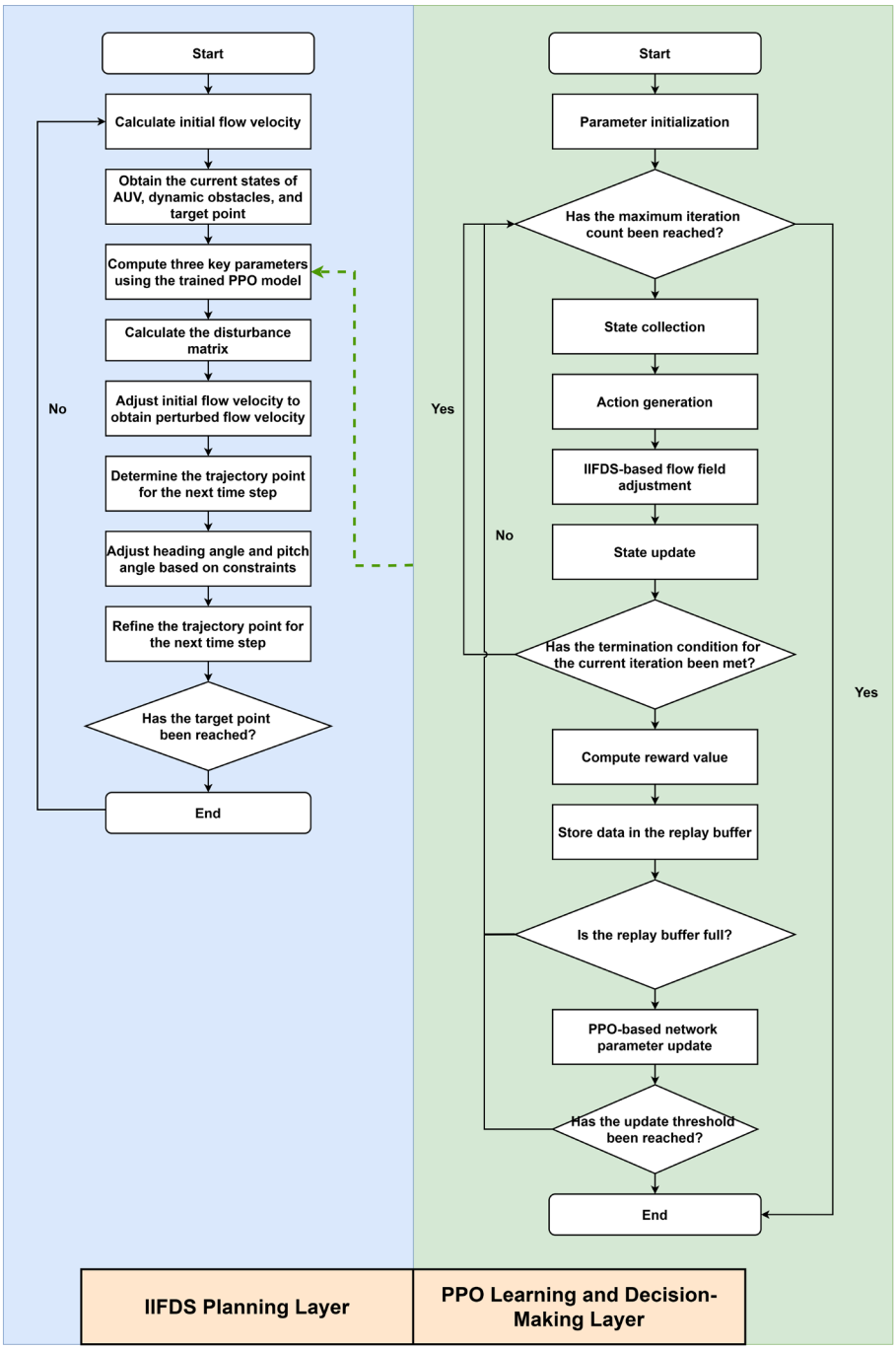

Figure 4. Framework of the trajectory planning method integrating PPO and IIFDS.

Initialization involves setting the IIFDS flow field parameters, constructing the AUV dynamics model and dynamic obstacle environment, and defining the start and target points. Simultaneously, the Actor–Critic network of the PPO algorithm is initialized, along with the replay buffer (size 4096). Multi-objective reward functions and related hyperparameters (discount factor, PPO clipping range, learning rate, etc.) are also configured.

Subsequently, the training loop is entered, which primarily includes the main loop and the PPO network update component.

The main loop constitutes the core part of the interaction between the AUV and the environment during training. The detailed steps of the main loop are as follows:

Step 1: State Collection

The AUV gathers environmental information through sensors to construct the current state, including the vector pointing to the target, $\mathbf{v}_{\text{goal}}$, and the vector pointing to the surface of the nearest obstacle, $\mathbf{v}_{\text{obs}}$, as well as the velocity vector of the nearest obstacle, $\mathbf{v}_{\text{obs}_{\text{s}}\text{peed}}$. The current state vector $\mathbf{s}_t$ is represented as follows:

$$\mathbf{s}_t = \left\{ \mathbf{v}_{\text{goal}}, \mathbf{v}_{\text{obs}}, \mathbf{v}_{\text{obs}_{\text{s}}\text{peed}} \right\} \tag{22}$$

Step 2: Action Generation

The action for the current state is generated through the Actor network of the PPO algorithm. First, the Actor network produces the parameters of a Gaussian distribution:

$$\mu, \sigma = \pi_\theta(s_t) \tag{23}$$

where μ represents the mean vector of the actions, and σ represents the standard deviation vector of the actions.

Subsequently, actions are sampled from the Gaussian distribution:

$$a_t \sim \mathcal{N}(\mu, \sigma), a_t = \{\rho_k, \sigma_k, \theta_k\} \tag{24}$$

Step 3: Flow Field Adjustment and Trajectory Planning

The planning layer of the IIFDS algorithm adjusts the flow field dynamically based on the action $a_t = \{\rho_k, \sigma_k, \theta_k\}$. The repulsion intensity of the obstacle is modified through ρ_k, the tangential effect range is adjusted through σ_k, and the flow field direction is altered via θ_k, enabling the AUV to avoid obstacles. The next trajectory point is planned using the following formula:

$$p_{t+1} = f_{\text{IIFDS}}(p_t, \rho_k, \sigma_k, \theta_k) \tag{25}$$

Subsequently, the AUV state is updated:

$$\mathbf{s}_{t+1} = \left\{ \mathbf{v}_{\text{goal}}, \mathbf{v}_{\text{obs}}, \mathbf{v}_{\text{obs}_{\text{s}}\text{peed}} \right\} \tag{26}$$

Step 4: Reward Calculation

Based on the current state $\mathbf{s}_t$, action a_t, and the next state $\mathbf{s}_{t+1}$, the instantaneous reward R_t is calculated as follows:

$$R_t = R_{\text{avoid}} + R_{\text{goal}} + R_{\text{smooth}} + R_{\text{kinematic}} + R_{\text{energy}} \tag{27}$$

Step 5: Data Storage

The current state, action, reward, and next state are stored in the experience replay buffer:

$$\text{Buffer} \leftarrow \{s_t, a_t, R_t, s_{t+1}\} \tag{28}$$

Step 6: Termination Check for the Main Loop

The main loop ends when any of the following termination conditions are met: The AUV reaches the target point. The AUV collides with an obstacle. The number of steps executed by the AUV reaches the predefined maximum value.

If none of the termination conditions are met, the process continues from Step 1 to execute the next step. If the termination conditions are met, the current episode ends, and a new episode begins by reinitializing the environment (including start point, target point, and obstacle information). Once the experience replay buffer is full (set to a capacity of 4096 in this study), it triggers an update of the PPO network.

First, the Critic network parameters θ_v are optimized using the mean square error loss function:

$$L_v(\theta_v) = \frac{1}{N}\sum_{i=1}^{N}(R_t + \gamma V(s_{t+1}) - V(s_t))^2 \tag{29}$$

where R_t represents the instantaneous reward, $V(s_t)$ represents the value function of the current state, and γ is the discount factor.

Next, the Actor network parameters θ_v are optimized using the objective function of PPO, which includes a clipping mechanism:

$$L^{\text{clip}}(\theta_\pi) = \mathbb{E}\left[min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}A_t, \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)A_t\right)\right] \tag{30}$$

where $A_t = R_t + \gamma V(s_{t+1}) - V(s_t)$ represents the advantage function, balancing the action a_t preference.

The training loop continues until the termination condition is met. The termination condition for the training loop is defined as follows: the variation in the parameters of the PPO's Actor and Critic networks falls below the predefined threshold, and the training iteration reaches the preset value.

During the testing process, the initial velocity of the AUV is first calculated, and based on the current state information, the pre-trained Actor network of PPO generates the updated IIFDS flow field parameters. Next, through optimized computation, the disturbance matrix is determined, and the velocity of the initial flow field is corrected to obtain the resultant velocity, which determines the trajectory point at the next time step. Then, based on the dynamics and kinematics constraints, this trajectory point is further corrected to obtain the final adjusted trajectory point at the next time step. Finally, whether the adjusted trajectory point reaches the target location is evaluated. If not, the loop continues. The above testing loop concludes when the target is reached, marking the end of the testing process.

3. Results

The framework for the three-dimensional dynamic trajectory planning of AUVs proposed in this paper is critical during the training phase. The most important aspect of training is the construction of a standardized simulated environment. Considering the uncertainty of dynamic obstacle motion in real-world tasks, the construction of the simulated environment introduces dynamic obstacles with varying motion speeds, radii, and trajectory changes. During training, each episode begins by randomly selecting an initial and a terminal point within the predefined range and then randomly selecting a dynamic obstacle from the set of predefined obstacles.

The main training settings are as follows: the maximum number of steps for AUV execution is set to 500; the learning rates of the Actor network and Critic network are both set to 0.0001; the replay buffer size is 4096; the batch size is 512; the number of repeated

training steps is 8; and the GAE advantage estimation parameter is 0.98. The training results are shown in Figure 5.

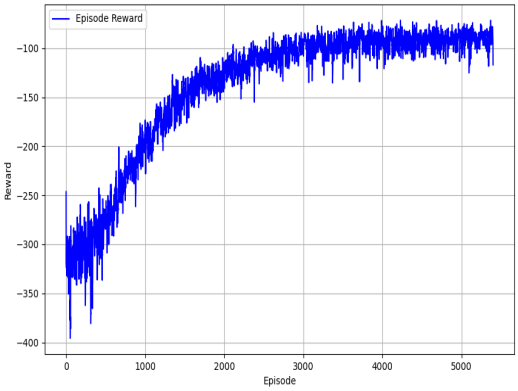

Figure 5. The reward function during training.

In Figure 5, the reward curve of the PPO-IIFDS framework illustrates the gradual optimization process from the initial exploratory strategies to the final effective strategies, demonstrating strong adaptability and robustness. During the initial phase of training, the reward rises rapidly, indicating that the model establishes its fundamental trajectory planning capabilities through interaction with the environment. In the middle phase, the reward growth slows down while the fluctuation amplitude decreases, reflecting the gradual improvement in the model's adaptability to random initialization and dynamic obstacle environments. In the later phase, the reward stabilizes, showing that the strategy has approached the globally optimal or near-optimal level, with minor fluctuations primarily caused by environmental randomness, exploratory actions, and multi-objective trade-offs. The statistical results further validate the high efficiency and robustness of the model, achieving high success rates (5361 successful tasks), low collision rates (39 failures), and near-zero superfluous stops. Overall, the reward's minor fluctuation demonstrates the rationality of the training process, reinforcing the model's ability to generalize in dynamic environments while confirming that the PPO-IIFDS framework effectively fulfills the task of three-dimensional dynamic trajectory planning.

3.1. Static Obstacle Environment Testing

In a static environment, we conducted tests on the IIFDS algorithm and the PPO-IIFDS framework, as shown in Figure 6. In the left panel of Figure 6, the start point is [0,10,10], the endpoint is [10,0,5.5], and the center coordinates of the static obstacle are [5,5,5.5]. In the right panel of Figure 6, the start point is [10,10,6], the endpoint is [0,1,3], and the center coordinates of the static obstacle are [6,6,5.5]. The influence range of the static obstacle is uniformly set to 2, and the repulsion coefficient ρ_k, tangential response coefficient σ_k, and directional coefficient θ_k of the IIFDS algorithm are fixed at 0.2, 0.2, and 0.1, respectively.

As shown in Figure 6, in static obstacle environments, the IIFDS algorithm with fixed parameters can plan relatively optimal paths in certain scenarios (as shown in the left panel). However, in other scenarios, it may result in paths passing too close to obstacles (as shown in the right panel). This is primarily because fixed parameters lack the flexibility required to adapt to different environmental features. In contrast, the PPO-IIFDS framework, through enhanced reinforcement learning, dynamically adjusts the repulsion coefficient ρ_k, tangential response coefficient σ_k, and directional coefficient θ_k, enabling it to generate more desirable trajectories in different scenarios. These trajectories effectively avoid obstacles while ensuring rationality and smoothness. The experimental results demonstrate that the PPO-IIFDS framework outperforms the traditional IIFDS algorithm in terms of robustness

and adaptability. This advantage allows the PPO-IIFDS framework to better accommodate diverse environmental characteristics and plan more efficient and safer trajectories, verifying its superior performance and potential for practical application in complex static obstacle environments.

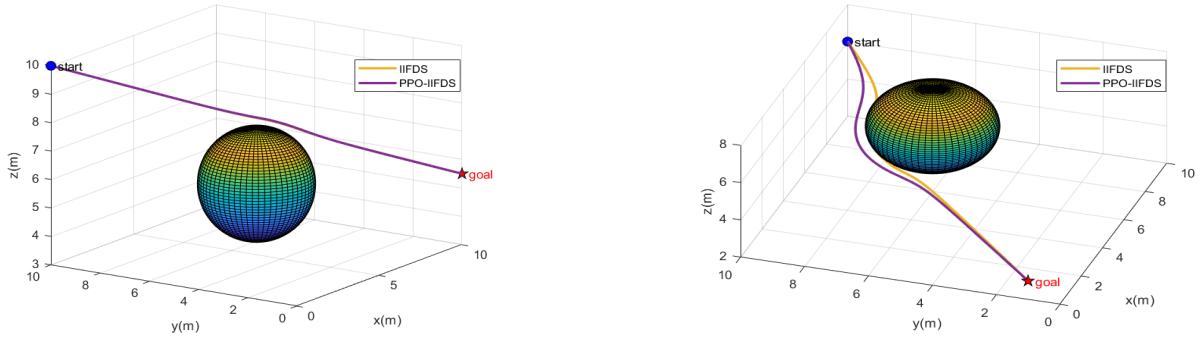

Figure 6. Comparison of IIFDS and PPO-IIFDS in different static environments.

3.2. Testing with Modified Reward Function

To perform a comparison with the results in Figure 5, we conducted experiments where all settings remained the same except for the exclusion of certain reward components, including various initialization parameters and the training environment. Specifically, this test only included obstacle avoidance rewards and target distance rewards. Figure 7 shows the reward changes during training under these conditions.

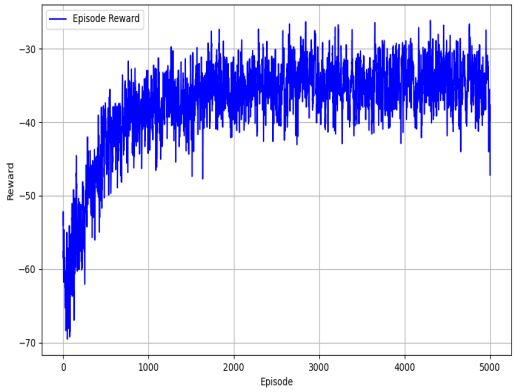

Figure 7. Reward function during training.

The reward variations in Figure 7 indicate that the model's learning performance was lower than that in Figure 5 when using only obstacle avoidance and target distance rewards.

From the reward curve in Figure 7, it can be observed that although the reward values show an upward trend during the initial phase of training, reflecting the model's gradual learning of obstacle avoidance and its mov toward the target, the growth rate of rewards slows significantly compared to Figure 5. Moreover, the reward fluctuations in the later phase are larger and less stable. This suggests that relying solely on obstacle avoidance and target distance rewards makes the model more prone to falling into local optima, resulting in less smooth trajectories and behavior that may fail to meet dynamics and kinematics constraints.

Next, we tested the trained models in two scenarios. For the single dynamic obstacle environment test, we first analyzed the movement of the dynamic obstacle.

We set the following reference position:

$$\text{obsref} = [x_{\text{ref}}, y_{\text{ref}}, z_{\text{ref}}] = [5, 8, 5] \tag{31}$$

The center position of the dynamic obstacle (obsCenter) changes over time. Its position in a three-dimensional space is defined by the following equations:

$$x(t) = x_{\text{ref}} + 3\sin(0.5t) \tag{32}$$

$$y(t) = y_{\text{ref}} + 3\cos(0.5t) \tag{33}$$

$$z(t) = z_{\text{ref}} + \sin(0.5t) \tag{34}$$

The velocity vector of the dynamic obstacle is the derivative of its position with respect to time, calculated as follows:

$$v_x(t) = \frac{\partial x(t)}{\partial t} = 1.5\cos(0.5t) \tag{35}$$

$$v_x(t) = \frac{\partial x(t)}{\partial t} = 1.5\cos(0.5t) \tag{36}$$

$$v_z(t) = \frac{\partial z(t)}{\partial t} = 0.5\cos(0.5t) \tag{37}$$

Based on the above equations, the movement of the dynamic obstacle exhibits the following characteristics:

Spatial trajectory characteristics: The obstacle moves periodically along a circular trajectory in the x-y plane with a radius of 3, while simultaneously performing small amplitude oscillations (with an amplitude of 1) in the z-direction.

Velocity characteristics: The magnitude and direction of the obstacle's velocity vary over time, governed by the sinusoidal and cosinusoidal functions. The velocity magnitude is determined by the equation $|\mathbf{v}(t)| = \sqrt{v_x^2(t) + v_y^2(t) + v_z^2(t)}$, and varies periodically with time.

As shown in Figures 8 and 9, Figure 8 presents the trajectory of the trained model under the single dynamic obstacle environment using the framework proposed in this study. Figure 9 shows the results when the trained model only uses obstacle avoidance and target distance rewards in the same environment. In the figures, the starting point is set to [0,2,5], and the target point is set to [10,10,5.5]. The blue circle represents the current position of the AUV, the green sphere represents the AUV's next position as calculated, the purple pentagram represents the target position, the yellow cuboid represents the dynamic obstacle, the red solid line represents the trajectory of the AUV, and the orange dashed line represents the trajectory of the dynamic obstacle.

Through Figures 8 and 9, it is evident that the model trained in Figure 9 lacks trajectory smoothness rewards, dynamics and kinematics constraint rewards, and energy efficiency rewards. As a result, the optimization process of the model primarily focuses on meeting the basic requirements of obstacle avoidance and reaching the target, while neglecting key indicators such as trajectory smoothness, physical constraints, and energy efficiency. In contrast, the comprehensive reward function design adopted in Figure 8 incorporates trajectory smoothness and dynamics constraint rewards, effectively guiding the model to achieve obstacle avoidance and target-reaching tasks while further enhancing the trajectory's smoothness and adaptability to dynamic environments. At the same time, the inclusion of energy efficiency rewards facilitates more energy-efficient trajectory planning. Thus, the results of Figures 8 and 9 further validate the comprehensiveness of the reward function design in the PPO-IIFDS framework. They also demonstrate that considering only a subset of reward terms significantly impacts the model's robustness and generalization ability. The comprehensive reward function design not only better aligns the

model with multi-objective requirements but also improves the global optimization level of trajectory planning.

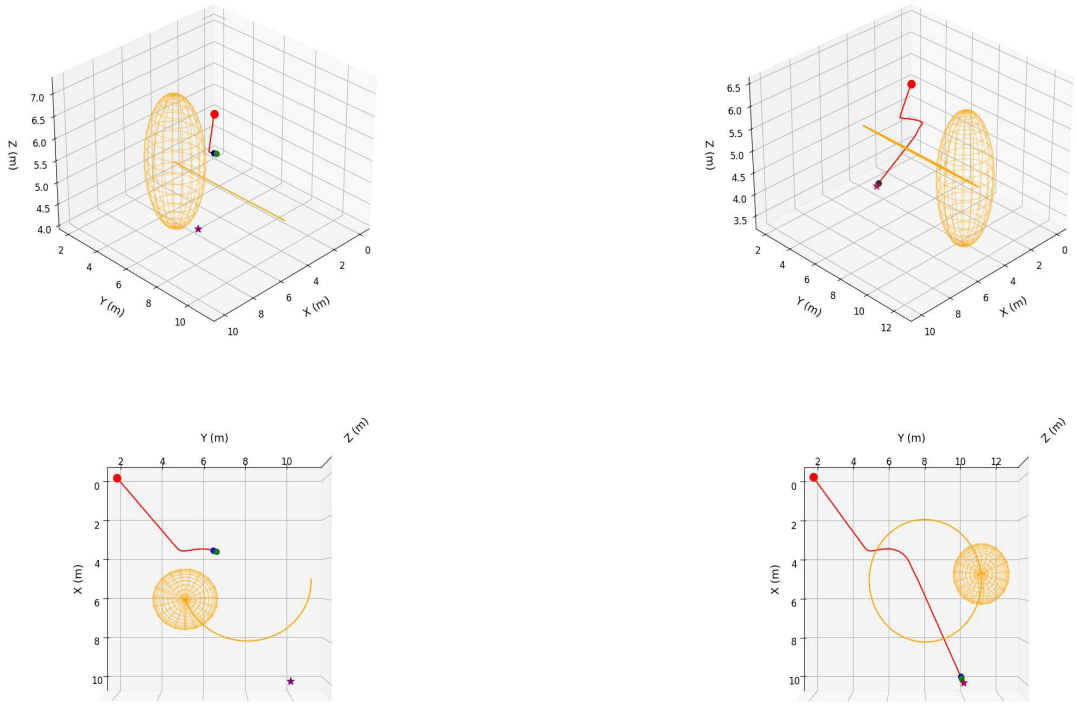

Figure 8. Results of the model trained using the framework proposed in this study in the single dynamic obstacle environment test.

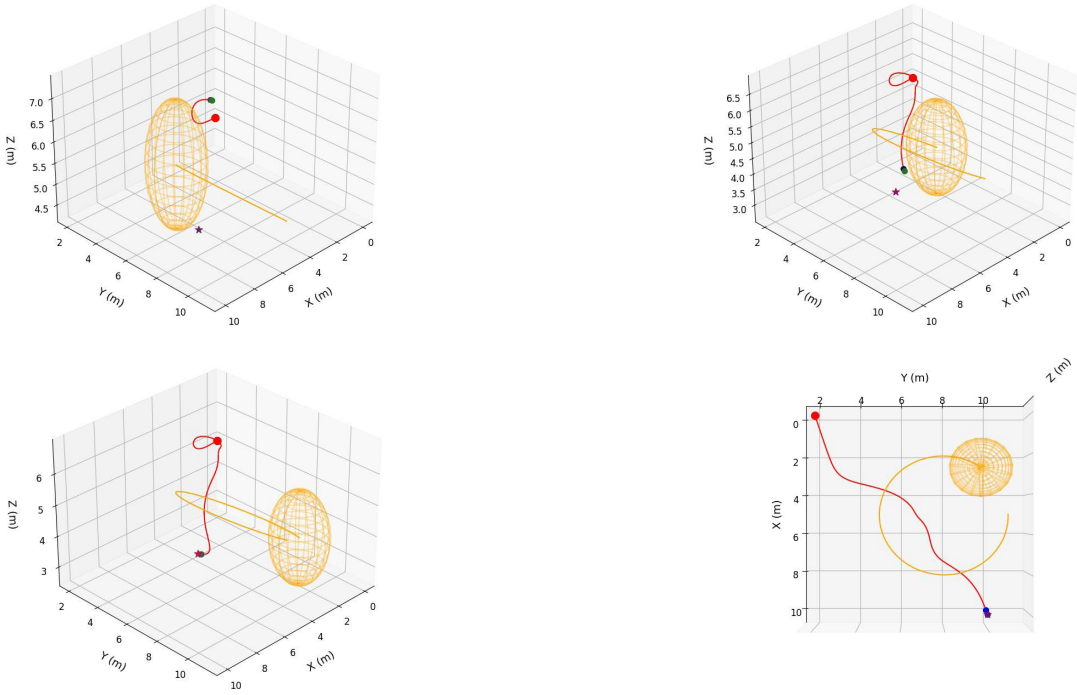

Figure 9. The results of the model trained with a reward function containing only obstacle avoidance and target distance rewards in the single dynamic obstacle environment test.

3.3. Testing in Complex Dynamic Obstacle Environments

To more comprehensively validate the robustness and generalization of the proposed PPO-IIFDS framework, we simulate a dynamic and complex underwater environment and test our model in an environment containing multiple dynamic obstacles.

First, we introduce the designed set of dynamic obstacles. These dynamic obstacle environments are intended to simulate the diverse characteristics of underwater dynamic obstacles, providing a variety of challenging scenarios for AUV 3D dynamic trajectory planning task training. The motion patterns of dynamic obstacles cover a range of dynamic characteristics, from simple to complex, including the following:

Circular motion (Dynamic Obstacle 1): The dynamic obstacle moves in a fixed-radius circle with a periodically changing speed, simulating underwater objects moving along a constant trajectory, such as underwater buoys or underwater work equipment.

Linear motion (Dynamic Obstacle 2, Dynamic Obstacle 3, Dynamic Obstacle 7): The dynamic obstacles move in a straight line with a constant or regularly changing speed. Dynamic Obstacle 2 and Dynamic Obstacle 3 are accompanied by single-axis oscillation, while Dynamic Obstacle 7 exhibits uniform drifting. These types of motion can simulate underwater carriers with uniform flow or obstacles that drift steadily.

Oscillatory motion (Dynamic Obstacle 4, Dynamic Obstacle 5, Dynamic Obstacle 6, Dynamic Obstacle 9): These dynamic obstacles exhibit complex periodic oscillations, covering both single-axis and multi-axis oscillations. For example, Dynamic Obstacle 4 combines planar circular motion with vertical oscillations, Dynamic Obstacle 5 demonstrates a combination of spiral and planar oscillations, Dynamic Obstacle 6 shows planar twisting oscillation characteristics, and Dynamic Obstacle 9 exhibits complex dynamic behavior through multi-axis oscillations. These movements simulate the behavior of obstacles influenced by underwater equipment operations or ocean current disturbances.

Spiral ascent motion (Dynamic Obstacle 10): This combines planar circular motion with axial progressive ascent, simulating the behavior of floating obstacles influenced by vortices or ascending bubble flows. The position and velocity equations for Dynamic Obstacle 10 are as follows:

$$x(t) = 6 + 2\sin(0.4t), y(t) = 6 + 2\cos(0.4t), z(t) = 4 + 0.5t \tag{38}$$

$$v_x(t) = 0.8\cos(0.4t), v_y(t) = -0.8\sin(0.4t), v_z(t) = 0.5 \tag{39}$$

From the formula, the motion characteristics can be seen: the trajectory exhibits spiral motion in the x-y plane, accompanied by uniform vertical ascent in the z direction. The velocity characteristic shows periodic changes in the horizontal velocity, while the vertical velocity remains constant.

Circular path retreat (Dynamic Obstacle 8): The dynamic obstacle initially moves along a circular trajectory, then gradually retreats to form a reciprocating motion, simulating the behavior of obstacles after force disturbances from underwater equipment or within a work area. The position and velocity equations for Dynamic Obstacle 8 are as follows, when $t < t_{\text{threshold}}$:

$$x(t) = 3 + 5\sin\left(\frac{\pi}{2} + 0.3t\right), y(t) = 10 + 5\cos\left(\frac{\pi}{2} + 0.3t\right), z(t) = 5 \tag{40}$$

$$v_x(t) = 1.5\cos\left(\frac{\pi}{2} + 0.3t\right), v_y(t) = -1.5\sin\left(\frac{\pi}{2} + 0.3t\right), v_z(t) = 0 \tag{41}$$

After exceeding the critical time, i.e., when $t > t_{\text{threshold}}$, the motion trajectory of Dynamic Obstacle 8 changes, entering a reverse regression phase. At this point, the formula is symmetrically adjusted, causing Dynamic Obstacle 8 to gradually return to its initial

state from the circular motion trajectory. From the formula, its motion characteristics can be observed: the trajectory is dominated by circular motion in the earlier phase and reverses in the later phase. The velocity characteristic shows that the velocity varies periodically with time.

These dynamic obstacle motion patterns are designed to replicate various dynamic obstacle characteristics that may be encountered in underwater dynamic environments, including floating devices, moving carriers, or dynamic objects affected by ocean currents. Through this diversified design, these environments comprehensively test the robustness and adaptability of the proposed algorithm in complex underwater dynamic environments, laying the foundation for the successful execution of real underwater tasks.

As previously mentioned, when the experience replay buffer is full (with a capacity set to 4096 in this paper), it triggers the update of the PPO network that we designed. Now, after updating the PPO network, we conduct tests in a multi-dynamic obstacle environment. First, we select four dynamic obstacles from the above dynamic obstacle set to form a dynamic obstacle combination environment. In total, we construct six dynamic obstacle combination environments, as shown in Figure 10:

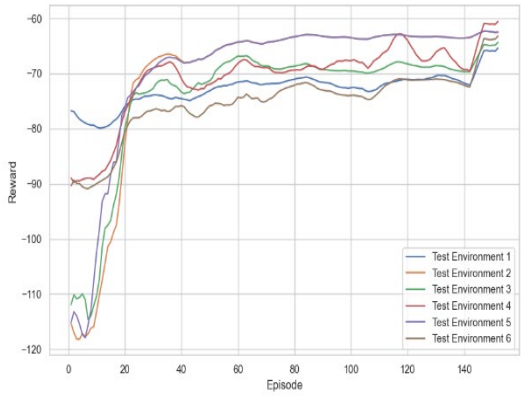

Figure 10. Testing results of the multi-dynamic obstacle environment after each update of the PPO network.

Figure 10 shows the testing performance of the PPO-IIFDS framework in a multi-dynamic obstacle combination environment. The reward function curve indicates that, with each update of the PPO network, the performance of the AUV gradually improves in each test environment, with the reward value converging to a higher level. This reflects the AUV's effective learning ability and adaptability in dynamic and complex scenarios. The reward curve changes in different environments show some variation. For example, in environments 2 and 5, the reward values stabilize quickly, indicating that the AUV can rapidly adapt and plan stable trajectories in these scenarios. However, in environments 4 and 6, there are some fluctuations, which may be due to increased environmental complexity, causing the AUV to require more time for exploration and optimization. Overall, the trend shows that the PPO network achieves good learning results in diversified dynamic obstacle combination environments, demonstrating excellent robustness and generalization abilities. This validates the effectiveness of the proposed method for trajectory planning in complex dynamic environments.

Finally, we test the trained model in the newly constructed dynamic obstacle combination environments. From the above dynamic obstacle set, we randomly select four dynamic obstacles to form a dynamic obstacle combination environment.

In Dynamic Obstacle Combination Environment A, we choose dynamic obstacles numbered [1,4,9,10] from the set. The starting point is set at [9,2,9], and the endpoint at [0,10,0]. The trajectory planning results using our trained model are shown in Figure 11. In

Dynamic Obstacle Combination Environment B, we select dynamic obstacles numbered [2,5,8,10] from the set. The starting point is set at [0,2,5], and the endpoint at [10,10,5.5]. The trajectory planning results using our trained model are shown in Figure 12.

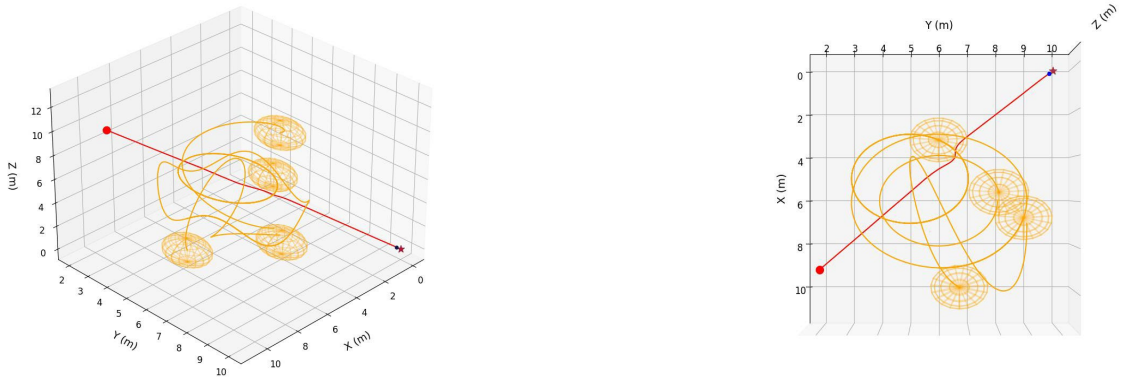

Figure 11. Testing results in Dynamic Obstacle Combination Environment A.

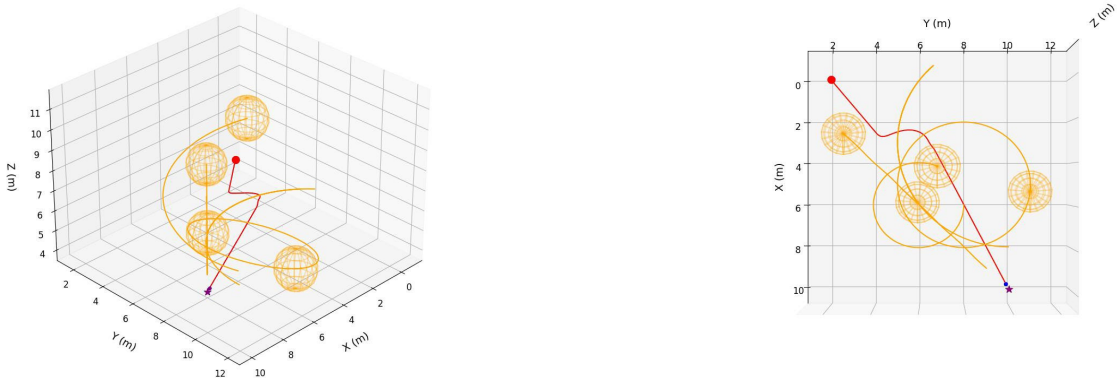

Figure 12. Testing results in Dynamic Obstacle Combination Environment B.

In the views, the blue sphere represents the current position of the AUV, the purple pentagram represents the target point, the red sphere represents the starting point, the orange spheres represent the dynamic obstacles, the red curve represents the AUV's trajectory, and the orange curve represents the trajectory of the dynamic obstacles.

Figures 11 and 12 show the trajectory planning results of the AUV in Dynamic Obstacle Combination Environments A and B, which validate the effectiveness of the trained model in complex dynamic environments. In Environment A, which contains dynamic obstacles numbered [1,4,9,10], the AUV starts at the point [9,2,9] and plans a collision-free path with a length of 15.17 m, completing the trajectory planning in 0.17 s. In Environment B, which contains dynamic obstacles numbered [2,5,8,10], the AUV starts at the point [0,2,5] and plans a collision-free path with a length of 13.96 m, completing the planning in 0.14 s. This demonstrates that the trained model can achieve efficient and reliable obstacle avoidance trajectory planning based on the trajectory characteristics of dynamic obstacles. Specifically, the AUV's trajectory can flexibly avoid dynamic obstacles, maintaining a safe distance from the obstacles, while planning the optimal path length between the target and the starting point. Additionally, the smoothness of the trajectory curve and the short planning time further verify the model's real-time performance and computational efficiency. These results indicate that the proposed algorithm can achieve robustness, generalization, and efficiency in various complex dynamic obstacle environments, meeting the demands of practical applications.

4. Discussion

The paper proposes a PPO-IIFDS framework for 3D dynamic trajectory planning in AUVs. The experimental results show that the PPO-IIFDS framework exhibits significant advantages in complex and dynamic obstacle environments. During training, the multi-objective reward function effectively guides the algorithm to optimize collision avoidance, target proximity, trajectory smoothness, dynamics constraints, and energy efficiency. In comparison, models trained with partial reward terms exhibit reduced optimization performance and efficacy, further validating the importance of comprehensive reward function design.

In static and dynamic obstacle environments, the PPO-IIFDS framework consistently demonstrates superior trajectory planning. Compared to the traditional IIFDS algorithm, the PPO-IIFDS framework produces smoother and safer trajectories while exhibiting strong adaptability to dynamic environments. Unlike traditional methods, which are limited by fixed parameter settings, PPO-IIFDS leverages reinforcement learning to dynamically adjust parameters such as the repulsion coefficients, tangential response coefficients, and directional coefficients. This adaptability enhances the trajectory quality and computational balance, addressing the traditional IIFDS algorithm's limitations in handling diverse scenarios.

Despite the progress achieved in this research, several directions merit further exploration. The primary focus of future work lies in the following areas:

(1) Theoretically, other continuous deep reinforcement learning methods can also be applied to the framework presented in this paper. Therefore, future work could integrate more advanced reinforcement learning algorithms, such as SAC [20] and TD3 [21], with the improved interfered fluid dynamic system (IIFDS) and conduct comparative tests with the approach proposed in this study.

(2) The PPO-IIFDS trajectory planning framework proposed in this study demonstrates strong robustness and adaptability, suggesting its potential for expansion into more complex autonomous underwater vehicle (AUV) task scenarios. Furthermore, we recommend conducting corresponding hardware experiments in real underwater environments to verify the feasibility and effectiveness of the algorithm in real-world tasks.

(3) The trajectory planning in this study utilizes a simplified AUV kinematic model and constraints, without incorporating more complex nonlinear dynamic models and controller characteristics. This could lead to an increased collision risk during execution due to controller delays or tracking errors. Therefore, future research could integrate planning, control, and dynamic modeling within the PPO-IIFDS framework to form a closed-loop system. By considering AUV dynamics and controller response characteristics in the reward function design, the reliability and safety of trajectory planning execution can be further enhanced.

(4) Future work will explore extending the PPO-IIFDS framework to UAV trajectory planning. Given the unique challenges UAVs face in dynamic obstacle environments, the framework's ability to dynamically adjust parameters (such as repulsion, tangential response, and directional coefficients) holds promise for effective obstacle avoidance and trajectory optimization. Future research will adapt the framework to UAV-specific needs and experimentally validate its feasibility and performance in complex aerial environments.

Author Contributions: Conceptualization, L.L. and M.S.; methodology, L.L.; software, M.S.; validation, L.L., M.S. and E.Z.; formal analysis, L.L.; investigation, E.Z.; resources, E.Z.; data curation, M.S.; writing—original draft preparation, L.L.; writing—review and editing, M.S.; visualization, K.Z.; supervision, L.L.; project administration, E.Z.; funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the projects of Zhejiang University of Science and Technology, under project numbers F701106N04 and F701106P03.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are not publicly available due to privacy restrictions and confidentiality concerns. Requests to access the datasets should be directed to the corresponding author at [222308855053@zust.edu.cn].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, D.; Wang, P.; Du, L. Path Planning Technologies for Autonomous Underwater Vehicles—A Review. *IEEE Access* **2019**, *7*, 9745–9768. [CrossRef]
2. Zhang, W.; Wang, N.X.; Wu, W.H. A Hybrid Path Planning Algorithm Considering AUV Dynamic Constraints Based on Improved A* Algorithm and APF Algorithm. *Ocean Eng.* **2023**, *285*, 115333. [CrossRef]
3. Ru, J.; Yu, H.; Liu, H.; Liu, J.; Zhang, X.; Xu, H. A Bounded Near-Bottom Cruise Trajectory Planning Algorithm for Underwater Vehicles. *J. Mar. Sci. Eng.* **2022**, *11*, 7. [CrossRef]
4. Ge, S.S.; Cui, Y.J. Dynamic Motion Planning for Mobile Robots Using Potential Field Method. *Auton. Robot.* **2002**, *13*, 207–222. [CrossRef]
5. Qiu, X.; Feng, C.; Shen, Y. Obstacle Avoidance Planning Combining Reinforcement Learning and RRT* Applied to Underwater Operations. In Proceedings of the OCEANS 2021: San Diego–Porto, San Diego, CA, USA, 20–23 September 2021; pp. 1–6. [CrossRef]
6. Huang, P.; Li, Y.; Wang, Y.; Guan, X. Information-Entropy-Based Trajectory Planning for AUV-Aided Network Localization: A Reinforcement Learning Approach. *IEEE Internet Things J.* **2025**, *12*, 2122–2134. [CrossRef]
7. Wang, F.; Zhao, L. Coordinated Trajectory Planning for Multiple Autonomous Underwater Vehicles: A Parallel Grey Wolf Optimizer. *J. Mar. Sci. Eng.* **2023**, *11*, 1720. [CrossRef]
8. Ge, S.S.; Fua, C.H. Queues and Artificial Potential Trenches for Multirobot Formations. *IEEE Trans. Robot.* **2005**, *21*, 646–656. [CrossRef]
9. Sullivan, J.; Waydo, S.; Campbell, M. Using Stream Functions for Complex Behavior and Path Generation. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Austin, TX, USA, 11–14 August 2003.
10. Waydo, S.; Murray, R.M. Vehicle Motion Planning Using Stream Functions. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003.
11. Liang, X.; Wang, H.; Li, D.; Lü, W. Three-Dimensional Path Planning for Unmanned Aerial Vehicles Based on Principles of Stream Avoiding Obstacles. *Acta Aeronaut. Astronaut. Sin.* **2013**, *34*, 1670–1681.
12. Wang, H.; Lü, W.; Yao, P.; Liang, X.; Liu, C. Three-Dimensional Path Planning for Unmanned Aerial Vehicle Based on Interfered Fluid Dynamical System. *Chin. J. Aeronaut.* **2015**, *28*, 229–239. [CrossRef]
13. Yao, P.; Wang, H.L. Three-Dimensional Path Planning for Unmanned Aerial Vehicles Based on an Improved Fluid Disturbance Algorithm and Grey Wolf Optimization. *Control Decis.* **2016**, *31*, 8.
14. Li, K.-W.; Zhang, T.; Wang, R.; Qin, W.-J.; He, H.-H.; Huang, H. Research Reviews of Combinatorial Optimization Methods Based on Deep Reinforcement Learning. *Acta Autom. Sin.* **2021**, *47*, 1001–1028.
15. Wang, D. Research Progress on Learning-Based Robust Adaptive Critic Control. *Acta Autom. Sin.* **2019**, *45*, 1031–1043.
16. Wang, D.; Ha, M.; Qiao, J. Self-Learning Optimal Regulation for Discrete-Time Nonlinear Systems under Event-Driven Formulation. *IEEE Trans. Autom. Control* **2019**, *65*, 1272–1279. [CrossRef]
17. Lyu, X.; Sun, Y.; Wang, L.; Tan, J.; Zhang, L. End-to-End AUV Local Motion Planning Method Based on Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2023**, *11*, 1796. [CrossRef]
18. Yuan, J.; Han, M.; Wang, H.; Zhong, B.; Gao, W.; Yu, D. AUV Collision Avoidance Planning Method Based on Deep Deterministic Policy Gradient. *J. Mar. Sci. Eng.* **2023**, *11*, 2258. [CrossRef]
19. Wu, J.; Wang, H.; Zhang, M.; Yu, Y. On Obstacle Avoidance Path Planning in Unknown 3D Environments: A Fluid-Based Framework. *ISA Trans.* **2021**, *111*, 249–264. [CrossRef] [PubMed]

20. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, New York, NY, USA, 10–15 July 2018; pp. 1861–1870.
21. Fujimoto, S.; Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. In Proceedings of the 35th International Conference on Machine Learning, New York, NY, USA, 10–15 July 2018; pp. 1587–1596.